Applications Services

Requirements Review

May 12, 1997

Version 1.0

1. Application Services

1.1 Application Services Introduction

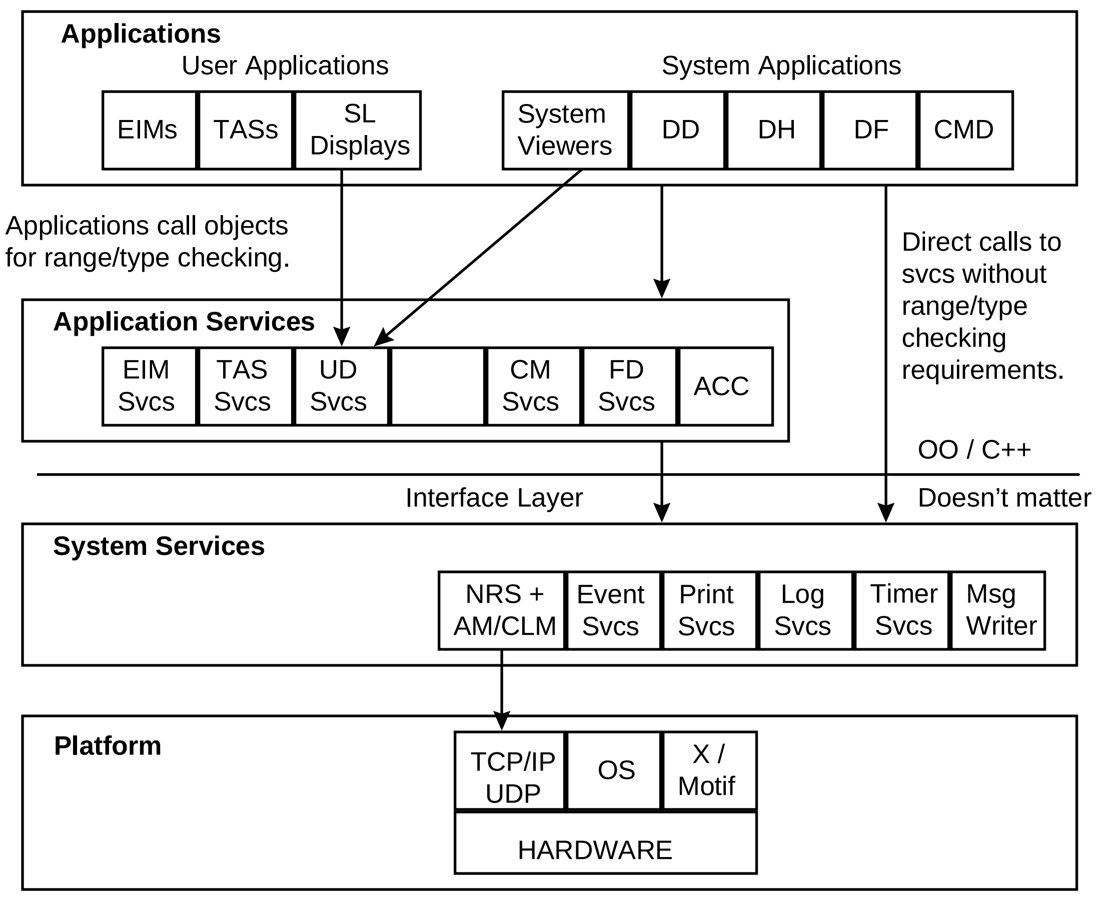
1.1.1 Application Services Overview

For Redstone, Application Services contains these CSCs:

- FD Services
- Constraint Management Services
- Application Communication and Control
- User Display Services

The following Application Services CSCs have no Redstone requirements:

- Data Path Services
- Data Fusion Services Specifications
- Math Model Services
- End Item Manager Services
- Prerequisite Control Services
- Reactive Control Services
- User Advisory Services Specifications
- System Advisory Services Specifications
- Test Application Script (TAS) Services Specifications



1.1.2 Application Services Operational Description

Application Services provides applications with an object-oriented interface to underlying services. Application Services implements a C++ layer that provides data type checking similar to the GOAL language.

1.2 Application Services Specifications

1.2.1 Application Services Ground-rules

1. All Application Services will provide or provide a transition to (by Thor) C++ APIs, using the approach agreed to with User Applications on 4/16/97.
2. Application Services will be implemented in C++ using single inheritance and minimal reference (pointer) usage to support possible future transition to Java.
3. Application Services will provide a method registration service for all application objects.

1.2.2 Applications Services Common Functional Requirements

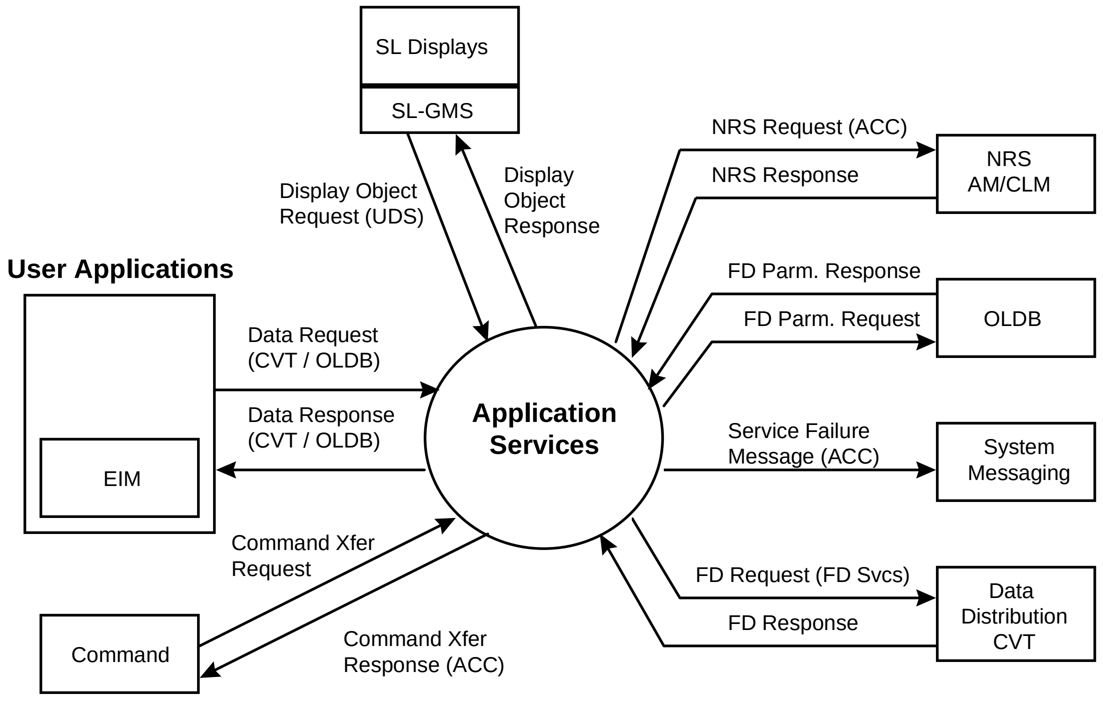
This section defines requirements common to all Application Services APIs.

1. *Application Services shall return status to the calling application on the success or failure of every API call made to an Application Service.*
2. *When an Application Services API call fails, the Application Service shall send an error message to the System Message Writer specifying the time and the reason (error code) for any failure condition.*

1.2.3 Application Services Performance Requirements

Redstone performance requirements are listed separately for each CSC.

1.2.4 Application Services Interfaces Data Flow Diagrams



2. Function Designator (FD) Services

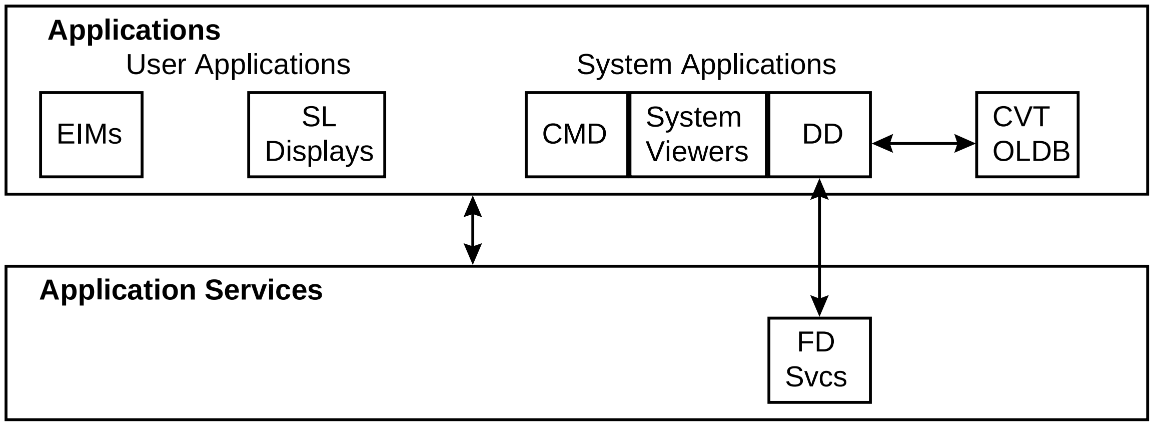
2.1 FD Services Introduction

2.1.1 FD Services Overview

"This service provides Function Designator (FD) measurement and stimulus data." FD Services shall perform the following Redstone services:

- Provide APIs allowing applications to access (raw & processed value) FDs.
- Provide APIs allowing applications to access queued multi-sample FDs.
- Provide APIs allowing applications to write (pseudo) FDs.

Function Designator (FD) Services provided applications a set of common access methods for reading and writing FD data values. These data values are stored in the OLDB, the Current Value Table (CVT), and the algorithm table (per C. King, 5/17/97).



2.1.2 FD Services Operational Description

2.2 FD Services Specifications

2.2.1 FD Services Ground-rules

Ground-rules for the Redstone delivery follow:

- *FD Services performs all FD reads and writes (including pseudo FDs) via Data Distribution.*
- The OLDB will be read-only for User Applications.
- There will only be one OLDB per "set".
- FD Services will use the OLDB provided by C. King, targeted for 7/7/97 availability.
- *The Redstone OLDB or Algorithm Table will contain Data Fusion algorithms and Data Health algorithms that will be located in the DDP. Viewers needing access to this data will retrieve it from the DDP via Application Services.*

2.2.2 FD Services Functional Requirements

The following paragraphs define the FD Services **Redstone** functional requirements:

- Read FDs
- Read Queued Multi-Sample FDs
- Write (pseudo) FDs.

Read FDs

1. FD Services shall provide an API to read the current data value for any valid FD.
2. FD Services shall provide an API to read the current value of an analog FD in engineering units for the analog FD types defined in KSC-LPS-OP-033-03, Part 1; GOAL User Guide.

3. FD Services shall provide an API to read the current value of a digital pattern FD.
4. FD Services shall provide an API to read the current value of a discrete FD for the following engineering unit types:
   a. Open / Close
   b. True / False
   c. Wet / Dry
   d. On / Off.
6. FD Services shall provide an API to read the current value of a discrete FD in raw format.
7. FD Services shall provide an API to read the time of the last change in value of the FD.
8. FD Services shall provide an API to read the health status the last change of an FD.
9. *FD Services shall provide the capability to access all current data attributes from the OLDB for any valid FD.*
10. *FD Services shall provide an API to read an FDs current value, time of last value change, and health status in a single request.*
11. *FD Services shall provide an API to read calibration data from the Gateways.*

Read Queued Multi-Sample FDs

1. *FD Services shall provide Queued multi-sample service to applications for any valid FD.*
2. *FD Services shall provide* an API *to identify that an FD that shall be delivered via queued service.*
3. *FD Services shall provide* an API *to access every change value in time synchronous fashion.*
4. FD Services shall provide an API to read the next value of a multi-sample FD.
5. FD Services shall provide an API to read the next N values of a multi-sample FD.
6. FD Services shall provide an API to clear all queued samples pending for the application.
7. *FD Services shall* provide an API to *notify user applications when multi-sample queued data is available for a relevant FD.*
8. *FD Services shall* provide an API to *start queued multi-sample delivery by FD.*
9. *FD Services shall* provide an API to *stop queued multi-sample delivery by FD.*

Write pseudo FD

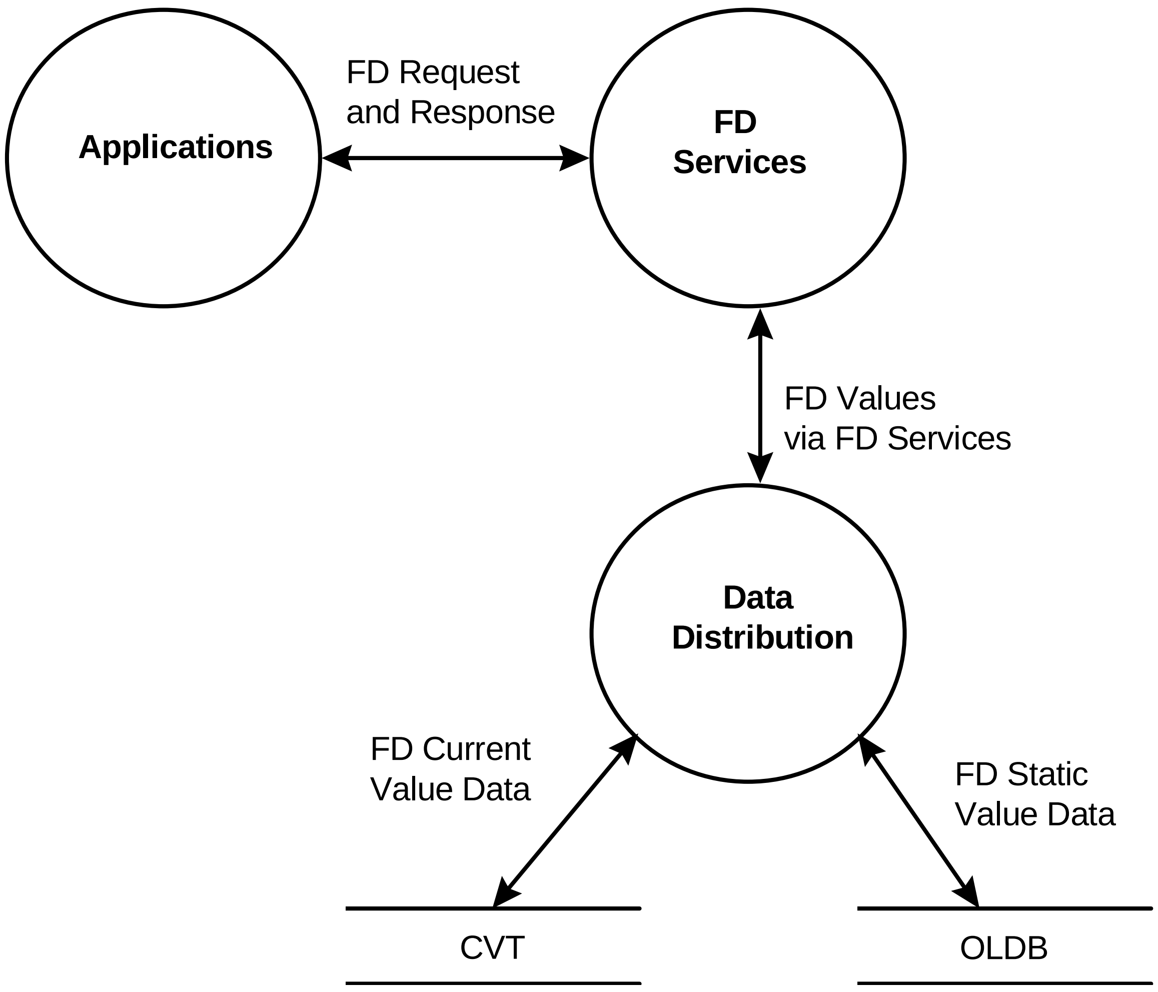
1. *FD Services shall provide an API to write values to analog output FD's.*
2. *FD Services shall provide an API to write a value to discrete output FD's.*
3. *FD Services shall provide an API to write discrete output FD's using the following discrete data types: OPEN, CLOSE, TRUE, FALSE, WET, DRY, ON, OFF.*
4. *FD Services shall provide an API to write a value to digital pattern output FD's.*
5. *FD Services shall provide an API to write a time value in TBD time format..*

2.2.3 FD Services Performance Requirements

- No specific performance requirements have been established for the FD Services Redstone delivery.
- Applications Services personnel will perform an FD Services study to determine the following:
  - CVT retrieval times (min, max, average) and quantities, by platform type
  - OLDB retrieval times (min, max, average) and quantities, by platform type
  - CVT retrieval times (min, max, average) and quantities, by platform type

Table	**Expected Accesses**
CVT	Multiple access per 10 ms
OLDB	Multiple access per second
Data Fusion / Data Health Algorithm Table	Multiple access per minute

2.2.4 FD Services Interfaces Data Flow Diagrams


Applications
FD Request and Response
FD Services
FD Values via FD Services
Data Distribution
FD Current Value Data
FD Static Value Data
CVT
OLDB

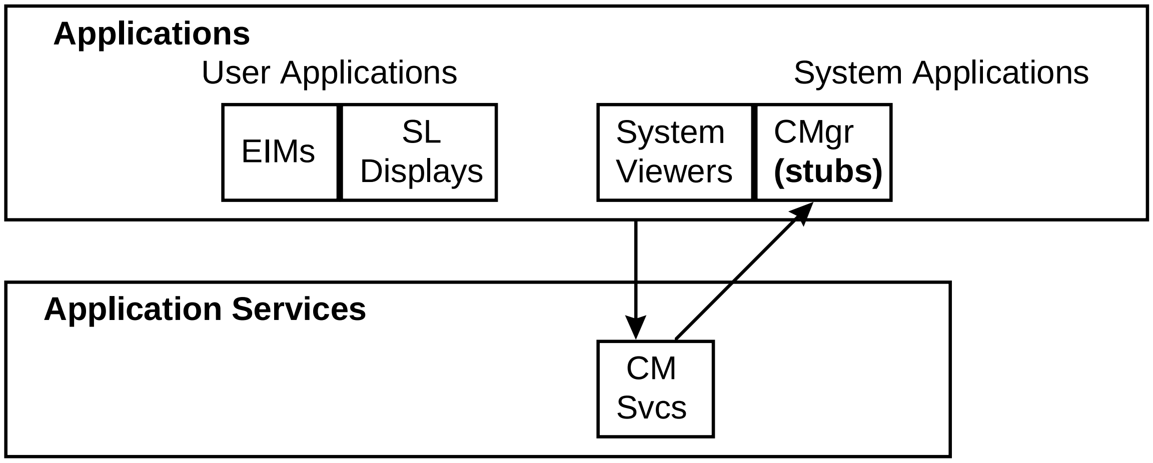
3. Constraint Management Services (CMS)

3.1 Constraint Management Services Introduction

3.1.1 Constraint Management Services Overview

This CSC provides the services required for applications to communicate with the Constraint Manager. Constraint Management Services provide applications a set of APIs for these functions:

1. *Activating and Inhibiting constraints*
2. *Changing constraints*
3. *Reading constraint information*



3.1.2 Constraint Management Services Operational Description

3.2 Constraint Management Services Specifications

3.2.1 Constraint Management Services Ground-rules

Ground-rules for the Redstone delivery follow:

- The Constraint Management CSC will not be available in the Redstone delivery.
- All CMS API interfaces to the Constraint Management CSC will be “stubbed” for Redstone.
- CMS APIs are lower priority than any other Redstone activities.

3.2.2 Constraint Management Services Common Functional Requirements

The following paragraphs define the FD Services **Redstone** requirements for these functions:

Activating and Inhibiting constraints

1. CMS shall provide an API to activate or inhibit constraint checking for an FD.

Changing constraints

2. CMS shall provide an API for changing the constraint limits associated with an analog FD.
3. CMS shall provide an API for changing the constraint state of a discrete FD.
4. CMS shall provide an API for changing the constraint condition for a digital pattern FD.

Reading constraint information

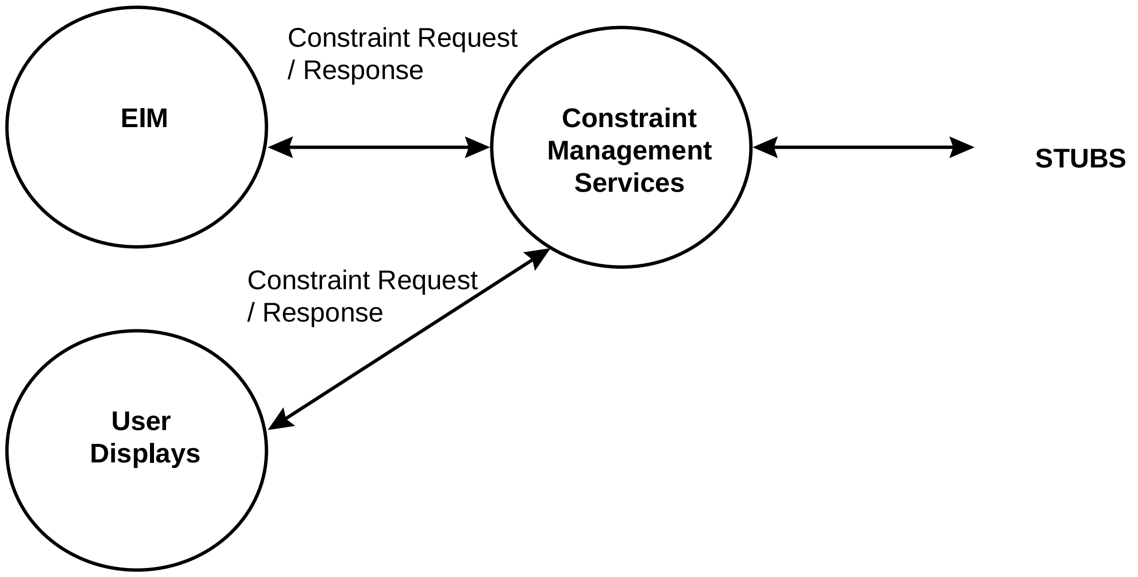
5. CMS shall provide an API for reading the constraint limits associated with an analog FD for an application.
6. CMS shall provide an API for reading the constraint state associated with a discrete FD for an application.
7. CMS shall provide an API for reading the constraint conditions associated with a digital pattern FD for an application.
8. *CMS shall provide an API for reading the time an FD violated a constraint.*

9. *CMS shall provide an API for reading the time an FD returned to limits.*

3.2.3 Constraint Management Services Performance Requirements

There are no Redstone performance requirements for Constraint Management Services.

3.2.4 Constraint Management Services Interfaces Data Flow Diagrams


Constraint Request / Response
EIM
Constraint Management Services
STUBS
Constraint Request / Response
User Displays

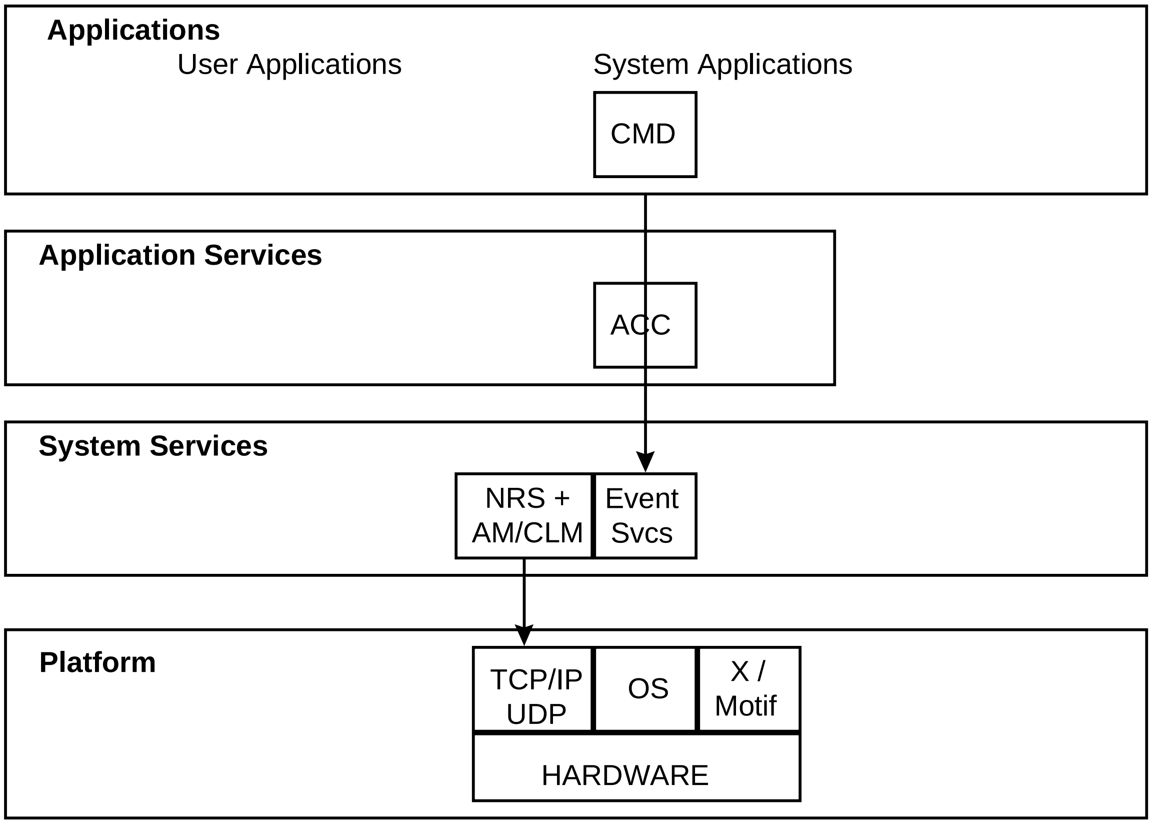
4. Application Communication and Control (ACC)

4.1 Application Communication and Control Introduction

4.1.1 Application Communication and Control Overview

"This CSC provides services needed for applications to communicate." Application Communication and Control performs the following for Redstone:

- Provides APIs allowing messaging between applications.
- Provides transparent intra-node and inter-node applications communications.
- Evaluates and recommends selection of an Application Communication and Control approach based on evaluation of various communications mechanisms (benchmark performance for various message sizes and traffic models; evaluate reliability/recoverability; determine labor/license costs; determine implementation impacts).
- Provides APIs for message separation and routing to support unique OLDB test set / flow zone configurations **(post-Redstone)**.



4.1.2 Application Communication and Control Operational Description

4.2 Application Communication and Control Specifications

4.2.1 Application Communication and Control Ground-rules

The Redstone ground-rules for Application Communication and Control Services follow:

- *ACC will use the existing Event Services software, provided by System Services.*
- ACC on the Gateway(s) will use AM/CLM and static NRS tables communicating to Gateway process(es) for applications.

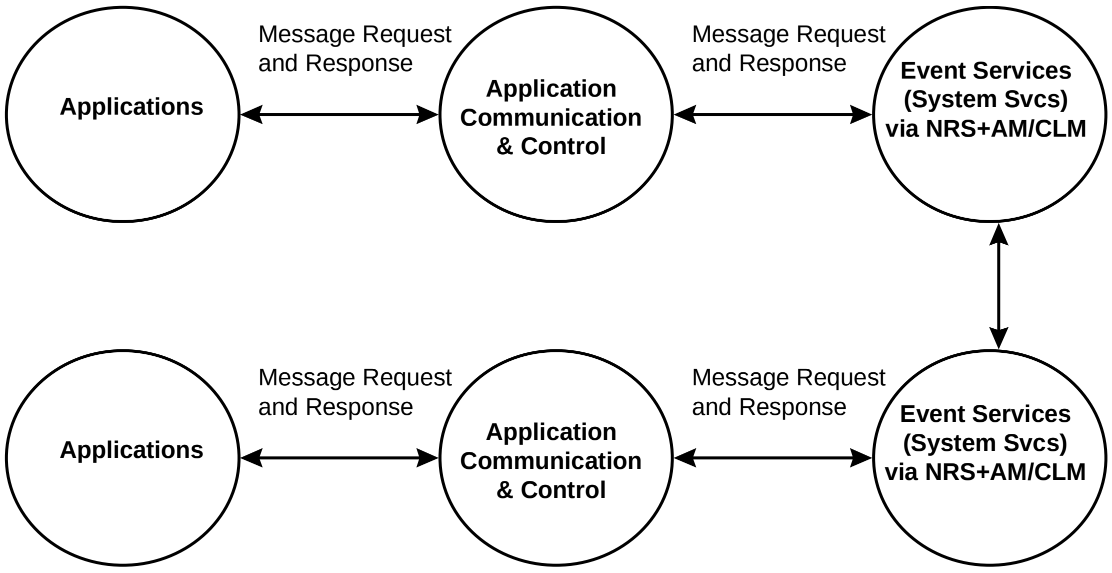
4.2.2 Application Communication and Control Functional Requirements

All Redstone ACC requirements are being implemented and tested under System Services.

4.2.3 Application Communication and Control Performance Requirements

- There are no specific Redstone performance requirements for Application Communication and Control.
- Application Services will perform the following performance analysis during Redstone:
  1. Benchmark C/non-OO (Event Services) vs. C++/OO (Object Mgmt Services) performance:
     - various message sizes (1, 10, 100, 1000, 10000 byte messages)
     - intra-node and inter-node
     - various traffic models ("bursty", continuous)
  2. Prototype / Evaluate CORBA Interface; re-run benchmarks.
  3. Recommend messaging mechanism.

4.4 Application Communication and Control Interfaces Data Flow Diagrams


Applications
Message Request and Response
Application Communication & Control
Message Request and Response
Event Services (System Svcs) via NRS+AM/CLM
Applications
Message Request and Response
Application Communication & Control
Message Request and Response
Event Services (System Svcs) via NRS+AM/CLM

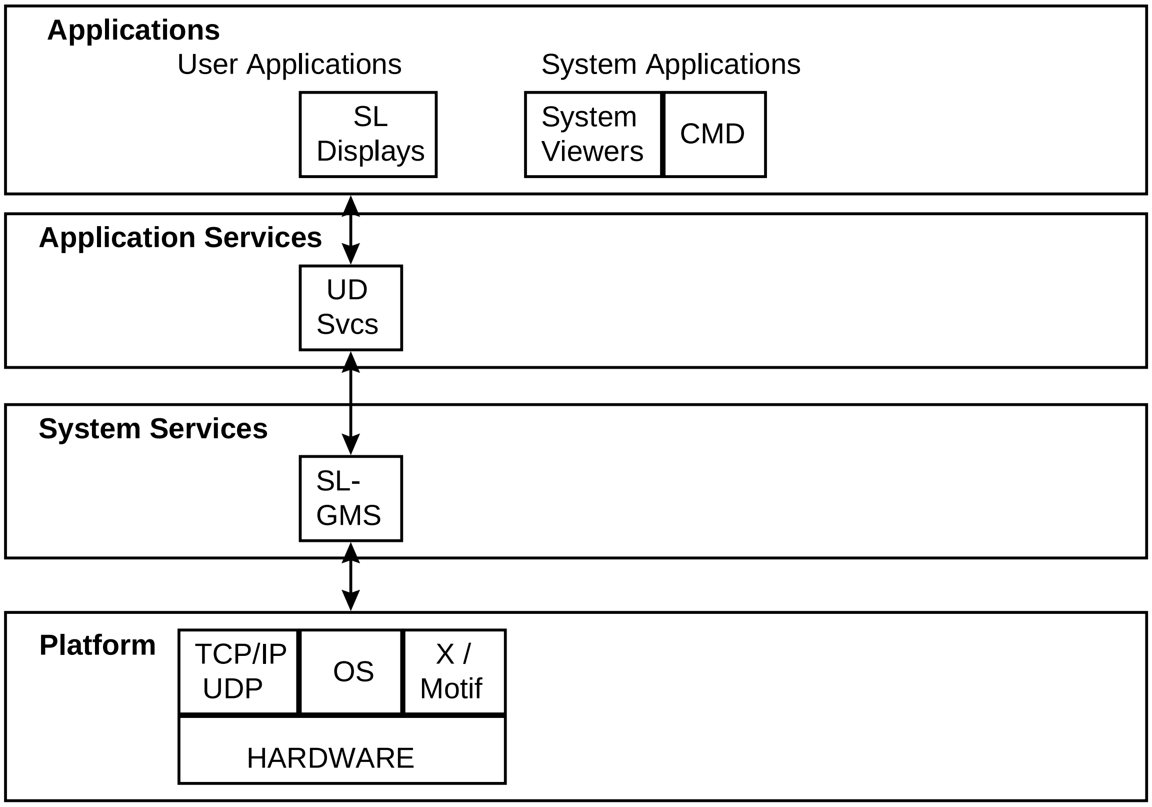
Application Communication and Control Interfaces Data Flow Diagram

5. User Display Services (UDS)

5.1 User Display Services Introduction

5.1.1 User Display Services Overview

“This CSC provides the capability for applications to communicate with displays.”
User Display Services are a series of common routines provided to application programs that provide visual displays.



5.1.2 User Display Services Operational Description

5.2 User Display Services Specifications

5.2.1 User Display Services Ground-rules

- UDS will interface to SL-GMS v.5.3 as the Redstone CLCS display development tool.
- All user displays will use User Display Services to access data (OLDB & CVT).

5.2.2 User Display Services Functional Requirements

The following paragraphs describe the **Redstone** UDS requirements:
1. UDS shall provide a mechanism allowing an FD-related display widget on an SL-GMS display to access to the corresponding FD data value or attribute (from the Current Value Table or the OLDB) without user (callback) programming.
System Viewer Services (for the Viewers).
2. UDS shall provide an API allowing System Viewers (FD, Constraint, Health, and Fusion Viewers) to access CVT and OLDB information.

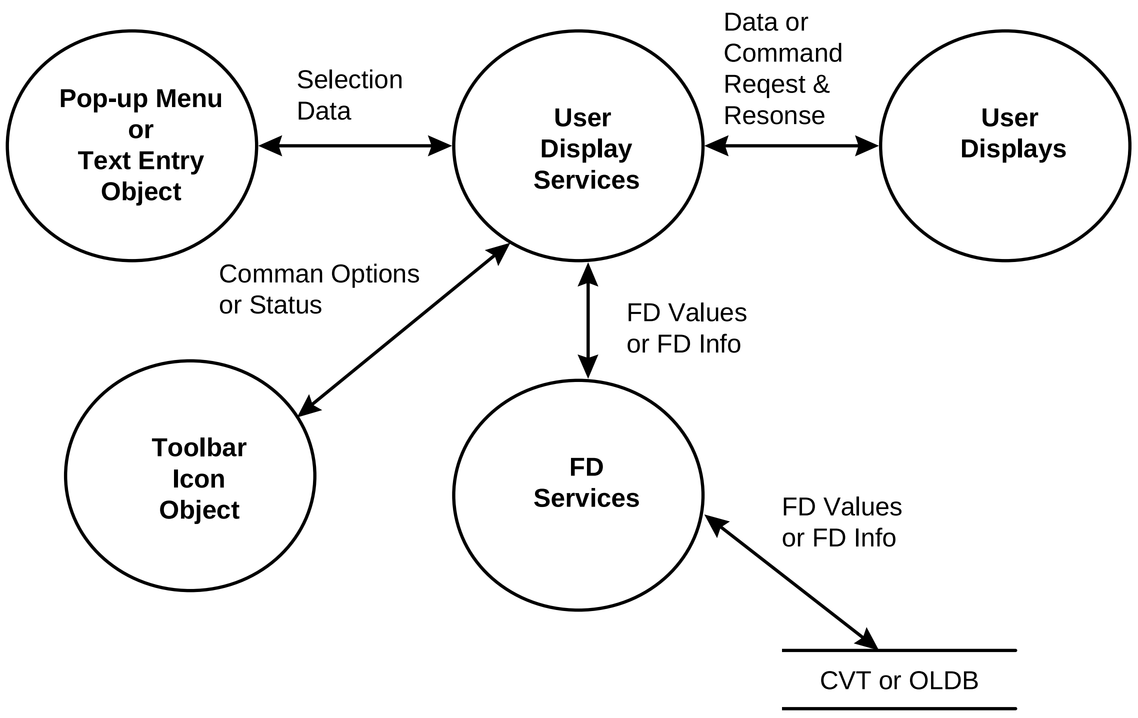
TBD Requirements:

1. UDS shall provide a set of color display attributes as follows:
   red = failed
   yellow = warning
   green = nominal
   white = invalid.
2. UDS shall provide APIs for SL-based Command displays to access objects.
3. UDS shall provide APIs for SL-based Command displays to access functions.
4. UDS shall provide APIs for SL-based Command displays to access pop-up text-entry displays:
   a. with associated menu selection items for available (valid) functions or options
   b. with configurable text length and title .
5. UDS shall provide APIs for SL-based displays to access pop-up displays for outputting Command-related data.
6. UDS shall provide a mechanism to attach a specified request (function) to mouse buttons as defined by System Viewers.
7. UDS shall provide a mechanism to display command options and / or status for each widget highlighted / selected by Command cursor position (a la MS Office tool bar icons).

5.3 User Display Services Performance Requirements

There are no specific Redstone User Display Services performance requirements.

5.4 User Display Services Interfaces Data Flow Diagrams


Pop-up Menu or Text Entry Object
Selection Data
User Display Services
Data or Command Reqest & Resonse
User Displays
Comman Options or Status
FD Values or FD Info
Toolbar Icon Object
FD Services
FD Values or FD Info
CVT or OLDB